SINGLE VIEW OBJECT COMPLETION

BY

JUSTIN JOSEPH THORSEN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Adviser:

Assistant Professor Derek Hoiem

ABSTRACT

The goal of this paper is to achieve complete reconstruction of a 3d object from a single depth image observation. Much effort has been put on multi-view reconstruction of objects with substantial success, but single view reconstruction is still very limited. Initial methods produce only partial reconstructions by projecting a depth map. State of the art approaches achieve complete reconstruction but either require user interaction or perform successfully on only a handful of simple categories. The method described in this paper is an exemplar based approach to fully automated reconstruction of a large variety of object classes using only a single depth image. The approach has three major components: retrieving a similar object, fitting the matched object to the query point cloud using alignment and symmetries, and reconstructing the mesh using the exemplar as a template. This method is evaluated in three distinct experiments: novel category (query of untrained class), novel model (query of trained class, untrained model), and novel view (query of trained model from a new viewpoint).

ACKNOWLEDGMENTS

This project could not have been completed without the support of many people. I would like to thank Jump Trading LLC for the extensive financial support that allowed my to pay my way through graduate school. I owe a great amount of gratitude Jason Rock, Tanmay Gupta, JunYoung Gwak, and DaeYun Shin for the intellectual contributions. I would also like to thank my adviser Derek Hoiem for the semesters of support, guidance, and encouragement. Lastly if it wasn't for the wonderful staff and equipment at the University of Illinois this project would never have completed due to the tens of thousands of compute hours, so a big thanks to the UIUC Vision Group.

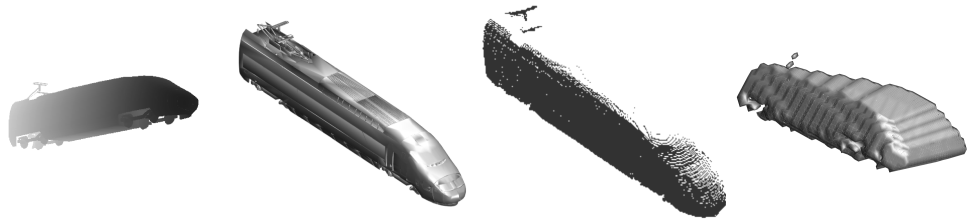
TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Overview

Consider the train in Figure 1. Although we only see a small portion of the object, we can accurately predict the entire shape. This ability to infer complete 3D shape from a single view is important for grasping, as we often reach around an object to grasp its unseen surfaces. Likewise, shape provides cues to category, affordance, and other properties. Recovery of 3D shape from a depth image is also useful for content creation and augmented reality applications. But how do we guess the shape of unseen surfaces? One approach is to recognize the same object or a similar object from past experience: the hidden surfaces of a favorite coffee mug can be inferred from earlier views or handling. Another is to infer missing surfaces using symmetries to duplicate and transform observed surfaces.



(a) Depth **(b)** True Mesh **(c)** Point Cloud **(d)** Reconstruction

Figure 1: While depth images and point clouds appear to have enough information for us to reconstruct a model, we are implicitly applying our intuition about what objects look like. For example we recognize that this is a cylindrical tram rather than a pyramid. (d) shows a reconstruction using a simple reflection about the image plane at maximum depth, which does not conform to our expectation. We aim to do better.

In this paper, we combine these strategies. Given a depth image, we find similar 3D models in similar viewpoints using view-specific matching of observed surfaces. The retrieved models are deformed to better approximate the visible portion of the query object. Deformations are constrained by smoothness and symmetries recovered from the 3D models to maintain a plausible shape. In some cases, such as when the instance or category is known, the deformed model will fit the observed surface, and hidden surfaces can be transferred from the model. In many cases, however, the deformed model does not fit the details of the observed surfaces, yet the query and retrieved object have similar symmetries. Thus, rather than transferring surfaces from the retrieved model, we instead transfer local symmetries to complete missing surfaces.

This paper is an extension of a previous work.[1]

1.2 Related Work

Most previous approaches in shape completion require user interaction (e.g. [29, 28, 10, 34, 5]) or are restricted to a limited set of categories (e.g., [2, 21], often chairs, as models are plentiful, and silhouettes informative. Our paper is unusual in its focus on quantitatively accurate reconstruction for a broad range of objects, rather than visually pleasing reconstruction for graphics content generation. We are the first to include a broad quantitative analysis of automatic 3D shape reconstruction of known objects from novel viewpoints, novel objects from known categories, and novel objects

[1]This project was submitted to CVPR 2015 [35] in collaboration with Jason Rock, Tanmay Gupta, JunYoung Gwak, DaeYun Shin, and Derek Hoiem. My main focus was on the silhouette matching baseline, the best voxel matching method, and the metrics and quantitative analysis of matching and deformations.

from unknown categories. Our approach includes techniques for viewpoint-based shape matching, 3D deformation, 3D mesh analysis, and 3D model synthesis that draw from or relate to a rich literature in graphics and computer vision.

Matching aims to find a complete 3D model from the library that has a similar shape to the depth-imaged object. A common approach, which we adopt, is to render each object from the library of meshes from many viewpoints and find a depth image that matches the query depth image to recover a similar 3D model from a similar viewpoint. For example, Ohbuchi et al. [25] proposes an approach for retrieval based on SIFT applied to depth images, which is used by Goldfeder et al.[11] to retrieve models in order to transfer grasp points. Wohlkinger and Vincze [31] describe a variety of matching features and an algorithm for partial-view to partial-view matching. Wang et al. [30] retrieve CAD models that are similar to a noisy 3D point cloud scan using local point descriptors and Regression Tree Fields. We experimented with many matching techniques but found that using Random Forests [4] to hash based on similarity and depth ordering of pixels provides a simple, fast, and effective retrieval mechanism.

Deformation aims to geometrically transform the retrieved mesh so that part of its surface aligns well with the observed query depth image. The key problems are finding local correspondences and defining a regularized deformation model that has sufficient flexibility but discourages unlikely shapes. Motivated in part by work in graphics on image-based modeling (e.g., Chen et al. [6]), we believe that symmetry-constrained deformations provide the right compromise between rigidity and arbitrary deformation. Our method for finding symmetries in the retrieved mesh is based on [23]. We use the Spin Image Descriptor [14] to find long-range correspondences for coarse, global (similarity) transformation, followed by nearest-point matching for iterative closest point [3] symmetry-constrained, thin plate spline deformation. Geometrically-constrained thin-plate spline models have been applied in various fields such as 2D graphics modeling [8] and medical imaging [1]. Kurz et al. [17] explore a generic constrained optimization framework for symmetry-preserving deformation that minimizes an energy function in a grid of B-spline basis functions. Other techniques that may be applicable include Sorkine and Alexa's As-Rigid-As-Possible deformation [28] and the cuboid-based deformations of Zheng et al. [34].

Completing a 3D model from an RGB image or depth image can be attempted by fitting a parameterized model based on contours, shape priors, and detected symmetries or by deforming an exemplar mesh. Contour-based methods often have limited applicability to specific classes, such as generalized cylinders (e.g., [22, 26]) or blocky structures [34], or require substantial user interaction (e.g., [6, 13]). Exemplar-based reconstruction typically requires finding very precise correspondences to very similar exemplar 3D meshes. Automatic and semi-automatic approaches exist for chairs [2] and furniture [21, 33] that have highly informative contours and an enormous supply of available 3D models. Kholgade et al. [16] propose a general exemplar-based completion approach that requires carefully chosen user correspondences. In other recent work, Wu et al. [32] present preliminary results on automatic shape completion from depth by classifying hidden voxels with a deep network.

CHAPTER 2

APPROACH

2.1 Overview

From an input segmented depth image, we wish to produce a complete 3D mesh. As we show in Figure 1, this is a highly challenging problem that requires recognition or strong priors about likely shapes, and simple methods such as reflecting the depth points along the camera axis are rarely effective.

Our approach, illustrated in Figure 2 is to first find a similar depth image from a training set of 3D meshes (Sec. 2.2). For retrieval, speed may trump clever similarity measures, because sub-linear methods enable querying into a large dataset of meshes rendered from many viewpoints. We use random forests as a hashing function to produce several candidates. The candidate with minimum surface-to-surface distance based on the query and retrieved depth images is selected to produce an exemplar 3D mesh and camera viewpoint.

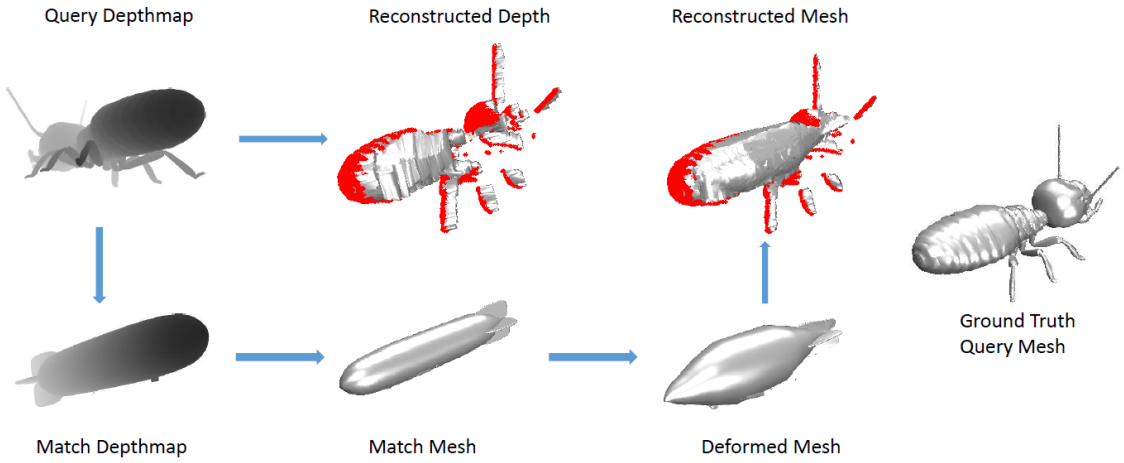


Figure 2: Pipeline for mesh reconstruction from depth image. A depth image of an ant is matched to a depth image of a blimp. The exemplar blimp is retrieved and deformed to better fit the observed depth points. Finally, a reconstructed mesh is created based on the observed depth points and deformed blimp exemplar. This mesh outperforms a mesh constructed from only depth points alone ("Reconstruct Depth").

The retrieved model may be of a different instance or category, and is likely to have a slightly different viewpoint. Therefore, we deform the retrieved mesh to better approximate the depth image (Sec. 2.3). Finding deformations to align different object models typically requires an experienced annotator to find corresponding points. We apply automatic methods to find likely correspondences between models and find symmetries within the exemplar 3D mesh that can be used as constraints to avoid unlikely warping. We then first apply a similarity deformation, followed by a symmetry-preserving thin-plate-spline deformation.

Our final step is to use the deformed exemplar mesh to complete the input depth image (Sec. 2.4). If the exemplar mesh is very similar to the query object, the deformed exemplar may provide a good estimate for the 3D shape. However, in many cases a similar object to the query will not exist in the dataset. For example, we may match an armored tank to a piano due to the similarity of their silhouettes and relative depth patterns from a particular viewpoint. Even in such cases, we show that the exemplar mesh can be very useful by predicting symmetries of the query object which are used to reflect observed points.

Then, a complete 3D mesh for the query object is constructed using graph cut to optimize a function that preserves observed and reflected depth points, while conforming to the exemplar mesh where the query and symmetries are uninformative.

2.2 Retrieving similar exemplar 3D meshes

Given a depth image of an object, we want to find a similar 3D mesh and viewpoint in our training set. In this discussion, we consider 3D shape to be view-specific because we wish to recover the 3D surface within the camera's reference frame. We render each training mesh from many viewpoints. The goal is to retrieve the training depth image such that the corresponding rotated 3D mesh will align with the partially observed 3D test object. Retrieval speed is important because the training set may contain many meshes rendered in many viewpoints.

Our approach is to use a random forest, trained to partition the training set into similar 3D shapes based on features of a depth image. Each tree of the forest acts as a hashing function, mapping the input features to a set of training examples. Random forests have been used for retrieval, e.g. by Fu and Qiu [9]. To retrieve based on shape similarity, we need to define entropy over a multi-dimensional variable (3D voxels), related to [7]. Our approach outperforms direct similarity-based matching and enables sub-second retrieval from a database of more than 65000 images.

Features Depth maps are first centered and resized to improve alignment of silhouettes. We use two simple types of features: whether a pixel at a given position is inside the silhouette and which of a pair of pixels is closer to the camera. The former encodes silhouette shape, and the latter encodes 3D shape in a manner that is scale invariant with respect to depth.

3D Shape Similarity Entropy Random forest training involves finding feature-based partitions of the data that reduce the entropy of a group of models. For each mesh viewpoint in the training set (depth image), we compute a 3D shape representation of 50^3. Since it is not feasible to calculate the entropy of a 50^3 dimensional feature, we take a low dimensional projection of the voxels using non-negative matrix factorization (NNMF) [18] with dimension 50, illustrated in Figure 3. We then discretize the NNMF coefficients into 3 values based on percentiles of non-zero values. We then compute the entropy of a set of shapes on the discretized coefficients, assuming that the coefficients are independent. For efficiency, we randomly subsample voxels before applying NNMF, which experiments indicate has little effect on accuracy while greatly improving speed. Then, we compute the coefficients of each new image using non-negative linear least-squares.

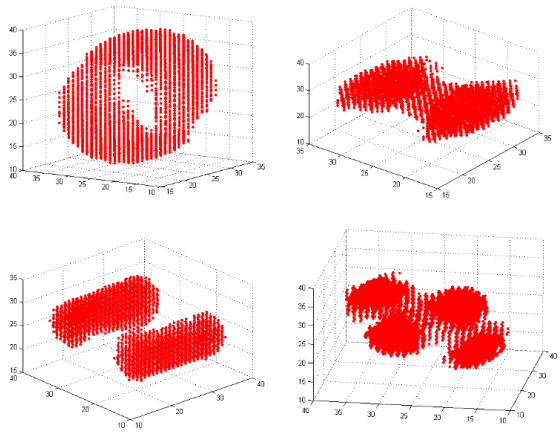


Figure 3: Low dimensional projection of voxels using non-negative matrix factorization.

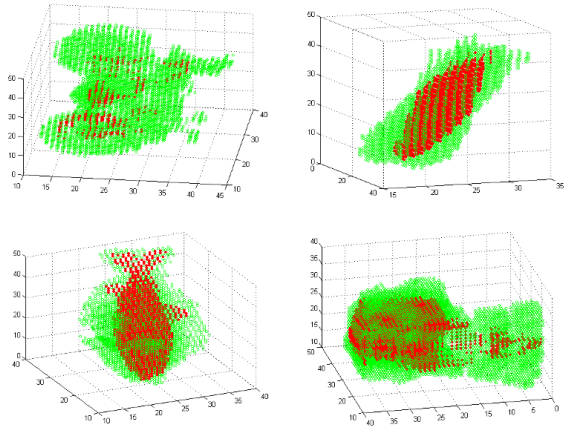


Figure 4: Voxel intersection(red) and union(green) of 3D shape examples in the same leaf node of our 3d-aware random forest. We can observe that similar objects with similar orientations are grouped into the same leaf node.

Training We train five trees, splitting until all leaf nodes contain five or fewer examples or cannot be improved with further splits. At each node, a random set of (binary) features is chosen (250 of each type), and the one that maximizes information gain is selected, with the requirement that the minimum number of images per node is $\max(.1S, 2)$, where S is the number of examples at the splitting node. The examples contained in leaf nodes are shown in Figure 4.

Retrieval Our random forest returns a set of potential matches. We could use each of these as a guess for the shape of the test object. For simplicity, we choose the match that has minimum surface-to-surface distance (Eq. 3.1) based on the query and retrieved depth images. Experimentally, we found that this selection procedure is nearly as effective as an oracle that selects the best potential match based on mesh-to-mesh distance.

2.3 Fitting retrieved exemplar to depth image

The exemplar retrieval process returns the 3D model in the training dataset with the most similar depth map to the query image. While the retrieved model is similar to the query, it still needs to be deformed to fit the query point cloud and better approximate the query shape. Here we describe our approach for symmetry constrained deformation for fitting a mesh to the query depth map.

Coarse Alignment We back project both the query and the retrieved depth maps in to 3D world coordinates using the camera parameters of the retrieved depth map. These two point clouds are aligned through a similarity transformation using spin-image [15] based correspondences. Spin images are a popular orientation invariant feature representation for 3D point clouds. Since we are matching 2.5D point clouds of different 3D models with similar but different orientations, there are a significant number of outliers. These outliers are first pruned by euclidean distance and then by a spectral correspondence matching technique [19] which considers a geometrically conforming set of correspondence pairs as the inliers. The similarity transform is obtained using Horn's closed form solution to the least square problem [12].

Symmetry Detection We find the significant symmetries in the retrieved mesh using [24]. We use a simplified version of this technique where we only consider planar symmetries and omit the symmetry basis reduction step. We also include a normal pruning step which is crucial in getting rid of the false symmetry matches. This is based on the fact that lines passing through the symmetry points should intersect at a point on the plane of symmetry. This technique produces a number of symmetry proposals from which we choose the top few based on the size of the mean-shift clusters.

While this gives us the major symmetry planes in the mesh, in order to incorporate local symmetries in our deformation model, we need to distribute these symmetries across the surface of the mesh. To do this we sample points uniformly randomly over the surface of the mesh. Then we reflect each point across every symmetry plane and match the reflection to the nearest sampled point. We retain only points within a threshold of the match distance.

Deformation We model the deformation using a 3D approximation thin-plate splines (TPS) model with additional symmetry constraints. The control points for the TPS model are chosen by uniformly sampling points on the surface of the mesh. This is in contrast to the usual TPS model where the source points are generally chosen as the control points. Note that we have correspondences on both sides of the symmetry planes which could potentially be far apart in 3D space. Thus our choice of control points somewhat decouples the learning of weights for points on either side of the symmetry plane, leading to better deformation behavior when used with symmetry constraints. We also replace the usual radial basis function ($r^2 log(r)$) with r^3.

The usual approximation TPS deformation model [27] regresses the parameters of a function $\mathbf{f}$ which maps every source point u to an approximation of their target location $\mathbf{v}$ within a certain tolerance to the bending energy involved. This is different from interpolation TPS where we require $\mathbf{f}(\mathbf{u}) = \mathbf{v}$ for all specified correspondence pairs $(\mathbf{u}, \mathbf{v})$.

However, in our symmetry constrained approximation TPS model we enforce symmetric points to deform in such a way that the symmetry is preserved. Thus given N correspondences $(\mathbf{u_i}, \mathbf{v_i})$, M symmetry pairs $(\mathbf{p_j}, \mathbf{q_j})$ and their

corresponding symmetry planes characterized by reflection mappings $\mathbf{R_j}$ such that $\mathbf{R_j}(\mathbf{p_j}) = \mathbf{q_j}$, we solve for a linear function $\mathbf{f}$ that minimizes the following objective

$$\sum_{i=1}^{N} \|\mathbf{v_i} - \mathbf{f}(\mathbf{u_i})\| + \lambda \mathcal{J}(\mathbf{f}) + \gamma \sum_{\mathbf{j=1}}^{\mathbf{M}} \|\mathbf{R_j}(\mathbf{f}(\mathbf{p_j})) - \mathbf{f}(\mathbf{q_j})\| \tag{2.1}$$

Here $\mathcal{J}$ is the bending energy involved in a given deformation and λ specifies our tolerance to this bending energy. $\lambda \to 0$ implies that we don't want to enforce any smoothness constraint while larger λ would result in a smoother deformation achieved by relaxing the correspondence mapping. The parameter γ enforces the strength of symmetry constraints. Since there could be a significant difference between the number of symmetry pairs and the number of correspondences, we use the following normalization to avoid bias

$$\gamma = \alpha \times \frac{Number\ of\ correspondences}{Number\ of\ symmetry\ pairs} \tag{2.2}$$

Now α weighs the strength of the symmetry constraints relative to the correspondence constraints. We have used $\alpha = 2$ and $\lambda = 0.001$ for all our experiments.

Note that finding parameters that minimize 2.1 is equivalent to finding the least square approximation to a set of linear equations. The linear equations for the usual approximation TPS model can be found in [27]. However, to minimize 2.1 we need to add an extra set of linear equations which are as follows

$$\mathbf{R_j}(\mathbf{f}(\mathbf{p_j})) - \mathbf{f}(\mathbf{q_j}) = \mathbf{0}\ \forall\ \mathbf{j} = \mathbf{1}, \cdots, \mathbf{M} \tag{2.3}$$

For a given symmetry plane defined by normal $\mathbf{n}$ and a point $\mathbf{m}$, the reflection mapping $\mathbf{R}$ is a generalization of the Householder transformation (which defines a reflection about a plane passing through the origin) to reflection about an arbitrary plane. Specifically, $\mathbf{R}$ is defined by a matrix-vector pair $(\mathcal{A}, \mathbf{t})$ where

$$\mathcal{A} = \mathbf{I} - \mathbf{2nn^T} \tag{2.4}$$

$$\mathbf{t} = \mathbf{2nn^T m} \tag{2.5}$$

Any point p can then be reflected about the plane by using the function $\mathbf{R}(\mathbf{p}) = \mathcal{A}\mathbf{p} + \mathbf{t}$

2.4 Completing missing surfaces

Finally, we need to construct a 3D mesh for our query object based on the observed depth points and deformed exemplar mesh. We pose this as a problem of predicting which voxels are occupied by the object. Voxels outside the depth silhouette or in front of observed depth points are known to be empty, while voxels around observed points are known to be occupied by the object. We then predict which unobserved voxels are occupied based on four types of cues.

Features The first two types of features are based on the observed depth map. First, voxels near observed depth points are likely to be occupied. $\mathbf{V}_{\text{dist}}(i)$ is the negative exponential of the distance of voxel i behind the depth map, in an orthographic projection. Voxels in front of the depth map have a distance of 1. Second, voxels that would require a large rotation to be exposed are more likely to be occupied (similar in concept to the accidental alignment principle). For example, a voxel behind a chair leg would have a higher accidental occlusion score than one behind the center of a wall. We compute $\mathbf{V}_{\text{ang}}(i)$ as the angle that the camera needs to move to make voxel i visible based on the contour of the depth map silhouette.

The third and fourth types of features are based on the retrieved and deformed mesh. We expect the query object

to have similar symmetries to the matched object. We reflect points on the query depth map using symmetries of the closest points from the matched mesh. $\mathbf{V}_{\text{Sym}}(i)$ is the negative exponential distance of voxel i in front of the symmetry-generated points. The final cue is that the query mesh should tend to have the same voxels occupied as the retrieved mesh. $\mathbf{V}_{\text{mesh}}(i)$ indicates whether voxel i is occupied in the deformed exemplar mesh.

Learning We train linear logistic regressors to predict which voxels are occupied. Our purely depth-based reconstruction uses only the first two feature types, while our retrieval-based reconstruction uses all four features.

Voxel Prediction The features and logistic regression models produce per-voxel unary terms, indicating the log ratio probability that the voxel is occupied. For voxels observed to be occupied or unoccupied, the unaries have a value of 5 added or subtracted to strongly encourage labels to fit observations. We then perform min cut optimization using a pairwise constant smoothing term of 10 in a 6-connected neighborhood in the 3D voxel space.

CHAPTER 3

EVALUATION

3.1 Datasets

We introduce a synthetic dataset built from the SHREC12 mesh classification dataset.[20] The SHREC dataset consists of twenty unaligned meshes from sixty diverse classes, including musical instruments, buildings, and vehicles. We align the meshes from each class consistently.

In Figure 5 we see properly aligned models. Each car model is in the same viewpoint as the others, and properly aligned to the ground plane. However the swords are poorly aligned. While they are all aligned properly to the ground plane, each varies slightly in its viewpoint. We generate 50 views rejection-sampled on a unit sphere bounded within 20 degrees of the equator. Alignment is used to generate sample viewpoints that avoid unlikely angles (e.g., looking at the bottom of a car), but alignment is not used in reconstruction.

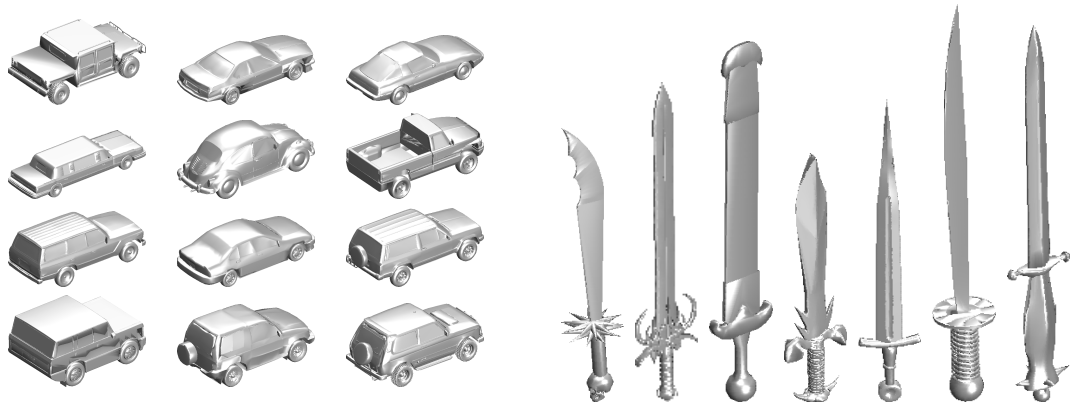


Figure 5: Manually aligned cars and swords. Some types of models such as cars are fairly easy to align to the ground plane and have a natural direction. Other models such as swords are more challenging and can have misalignment in certain areas such as the blade direction or hand guards.

We consider three types of reconstruction problems:

- *Novel View:* An object is seen from a new viewpoint, but the 3D model of the object is present in the dataset. For each model in the dataset, we split views into training, validation, and testing.
- *Novel Model:* The target object is from a known category, but the instance is new. For example, the input is a depth image of a car that is not present in the exemplar dataset, but models of other cars are present. For each category, we use 10 instances for training, 5 for validation, and 5 for testing.
- *Novel Category:* The target object is from an unknown category. For example, the target object is a bi-plane, and there are no airplanes in the exemplar set. We use thirty classes for training, 15 for validation, and 15 for testing.

In total, we have 600 examples each for testing novel model and novel category, and 900 examples to test novel view. Note that in testing, it is not known to the algorithm whether the corresponding model or category of a viewed object is present in the exemplar set.

3.2 Metrics

For evaluating our methods, we propose two metrics. The first is the intersection over union for two voxels (V_{iou}). We construct a voxel map of 200^3 voxels for both our query and match, and compute the intersection over union in 3D space. We analyze match and shape consistency using a surface distance metric by sampling points on the meshes and computing the normalized point cloud to point cloud distance. This gives us an estimate of the surface agreement of our meshes.

$$V_{pcl} = \text{mean} \min_m \|q - m\| + \min_q \|m - q\| \tag{3.1}$$

For each of the random forest methods we retrieve a set of potential matches. We compute our surface distance metric between the query depth point cloud and each of the potential match point clouds. Since our random forest trains to minimize voxel entropy, using the surface distance as a decision metric between potential matches allows us to achieve strong results for both metrics. In Table 1, we compare our selected potential match to the best match and median match for each of the metrics. On average our selection method chooses a match whose metrics are better than 88% of the potential matches.

Voxel I/U		Best Match	Median Match	Selected Match	Percentile
Novel Class	Mean	0.246	0.121	0.199	78
	Median	0.217	0.086	0.151	90
Novel Model	Mean	0.368	0.186	0.325	84
	Median	0.333	0.141	0.264	90
Novel View	Mean	0.502	0.263	0.478	90
	Median	0.507	0.196	0.482	100
Surface Distance		Best Match	Median Match	Selected Match	Percentile
Novel Class	Mean	0.058	0.087	0.063	90
	Median	0.056	0.084	0.060	100
Novel Model	Mean	0.036	0.063	0.039	92
	Median	0.032	0.059	0.033	100
Novel View	Mean	0.024	0.050	0.025	95
	Median	0.017	0.046	0.018	100

Table 1: Our selected potential match is, on average, in the top 22% of potential matches.

In Figure 6, we show a qualitative comparison of matches with different metrics. We show matches with scores in the best 10% for both metrics, matches with surface distances in the best 25% and voxel scores in the worst 25%, matches with voxel scores in the best 25% and surface distance scores in the worst 25%, and finally matches with scores in the worst 10% for both metrics. When both metrics score high, the match is generally a good, and when both are low the match is poor. When a voxel score is poor but the surface distance score is good the match is also fairly good. Surface distance also captures surface details, while an average voxel score captures larger structures. For example in Figure 6 the center right match (the fish and helicopter) was a query of known class that matched out of class. Due to the large central structures of each mesh we achieved high a voxel score but certainly not the optimal match in the database.

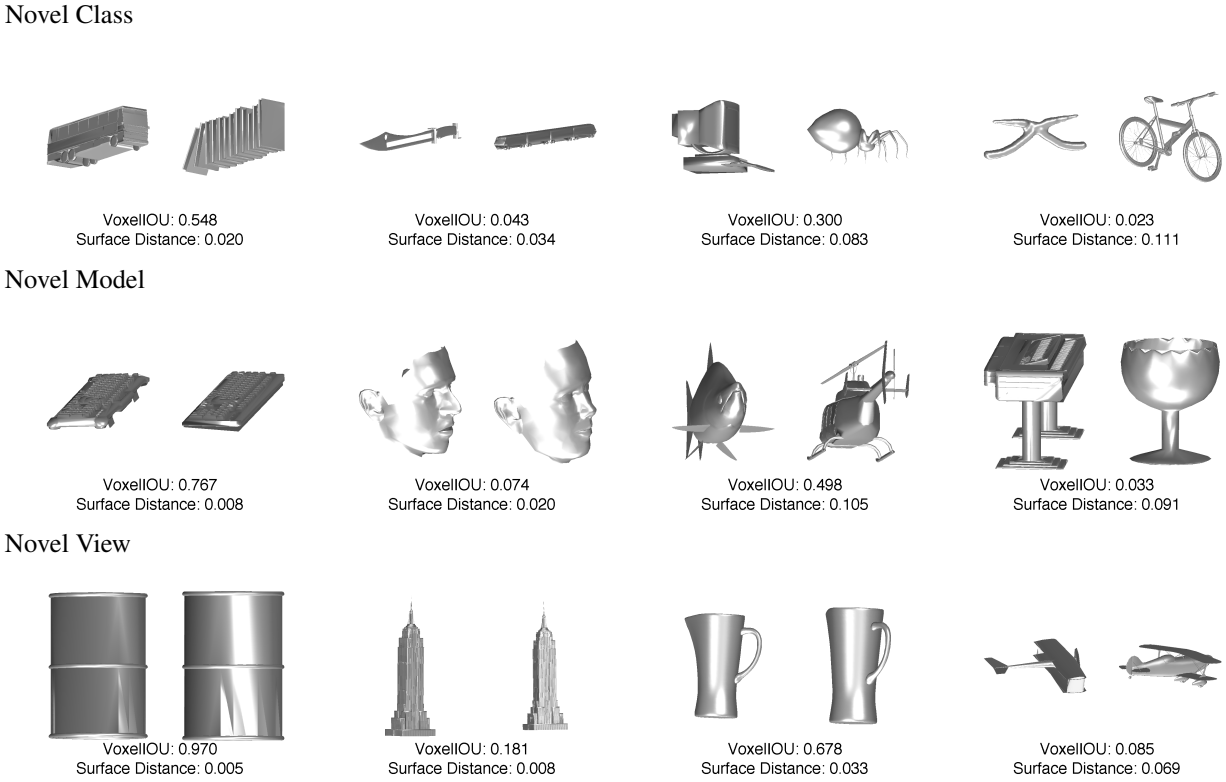


Figure 6: Qualitative comparison of match results. From left to right: matches with scores in the best 10% for both metrics, matches scores in the best 25% for surface distance and worst 25% for voxel I/U, matches with scores in the best 25% for voxel I/U and worst 25% for surface distance, matches with scores in the worst 10% for both metrics.

3.3 Experiments

Shape retrieval In Table 2, we compare similarity of the retrieved and query mesh before deformation. We compare retrieval methods of brute force silhouette matching (based on intersection over union of depth map silhouettes), matching using completely random forests (hashing using random forest structures without training based on shape entropy) using silhouette features or both silhouette and depth features, and our described learned random forest retrieval ("RF 3D Entropy"). Our proposed method achieves better surface distance scores in all experiments, and better voxel scores for novel class and novel model experiments. We perform similarly to other methods in terms of voxel scores for the novel view experiment. As expected, the similarity of the retrieved mesh is greatest for "novel view", next greatest for "novel model", and least for "novel category".

Voxel I/U		Silhouette	RF Silhouette	RF Sil+Depth	RF 3D Entropy	Best Voxel Match
Novel Class	Mean	0.196	0.194	0.199	0.199	0.389
	Median	0.147	0.143	0.147	0.151	0.401
Novel Model	Mean	0.305	0.292	0.308	0.325	0.482
	Median	0.241	0.233	0.252	0.264	0.486
Novel View	Mean	0.482	0.446	0.445	0.478	0.592
	Median	0.506	0.413	0.423	0.482	0.632
Surface Distance		Silhouette	RF Silhouette	RF Sil+Depth	RF 3D Entropy	Best Voxel Match
Novel Class	Mean	0.072	0.066	0.065	0.063	0.057
	Median	0.065	0.063	0.061	0.060	0.051
Novel Model	Mean	0.047	0.043	0.042	0.039	0.036
	Median	0.040	0.040	0.037	0.033	0.027
Novel View	Mean	0.029	0.029	0.029	0.025	0.018
	Median	0.016	0.023	0.023	0.018	0.013

Table 2: The above table shows how each matching method performs for each of three experiments. We evaluate our methods using our two metrics: voxel intersection over union and point cloud surface distance. The matching methods above are from left to right: brute force silhouette matching baseline, random forest matching with silhouette features, random forest matching with silhouette and depth features, random forest matching with depth features and voxel entropy minimization, brute force voxel intersection over union matching.

We have shown in Figure 6 that our metrics are reasonable, that they correlate to good qualitative matches under the right conditions and that our matching methods produce reasonable results on average. In Figure 7 we display histograms of matches for each experiment and metric. This shows that a reasonable amount of matches have good scores. Empirically from Figure 6 we determine the cut off of "good" matches to be within these metric bounds:

$$V_{pcl} < .05 \cup (V_{pcl} < .08 \cap V_{iou} > .3) \cup (V_{pcl} < .12 \cap V_{iou} > .5) \tag{3.2}$$

Given this condition we plot V_{you} against V_{pcl} in Figure 7 and color good matches blue. As expected for novel class we have the fewest good matches (44%), increasing good matches for novel model (76%), and the highest percentage of good matches for novel view (89%).

Shape completion We compare our reconstruction method with several baselines and intermediate results in Table 3. "Reflect Depth" is computed by reflecting the observed depth points across a plane parallel to the image plane at maximum depth. It performs very poorly. "Reconstruct Depth" uses only the query depth map and the first two feature types to predict voxel occupancy. "Match" shows reconstruction accuracy proposing the retrieved 3d exemplar mesh directly as the query shape, "Aligned" is after similarity transform, and "Deformed" after symmetric-preserving

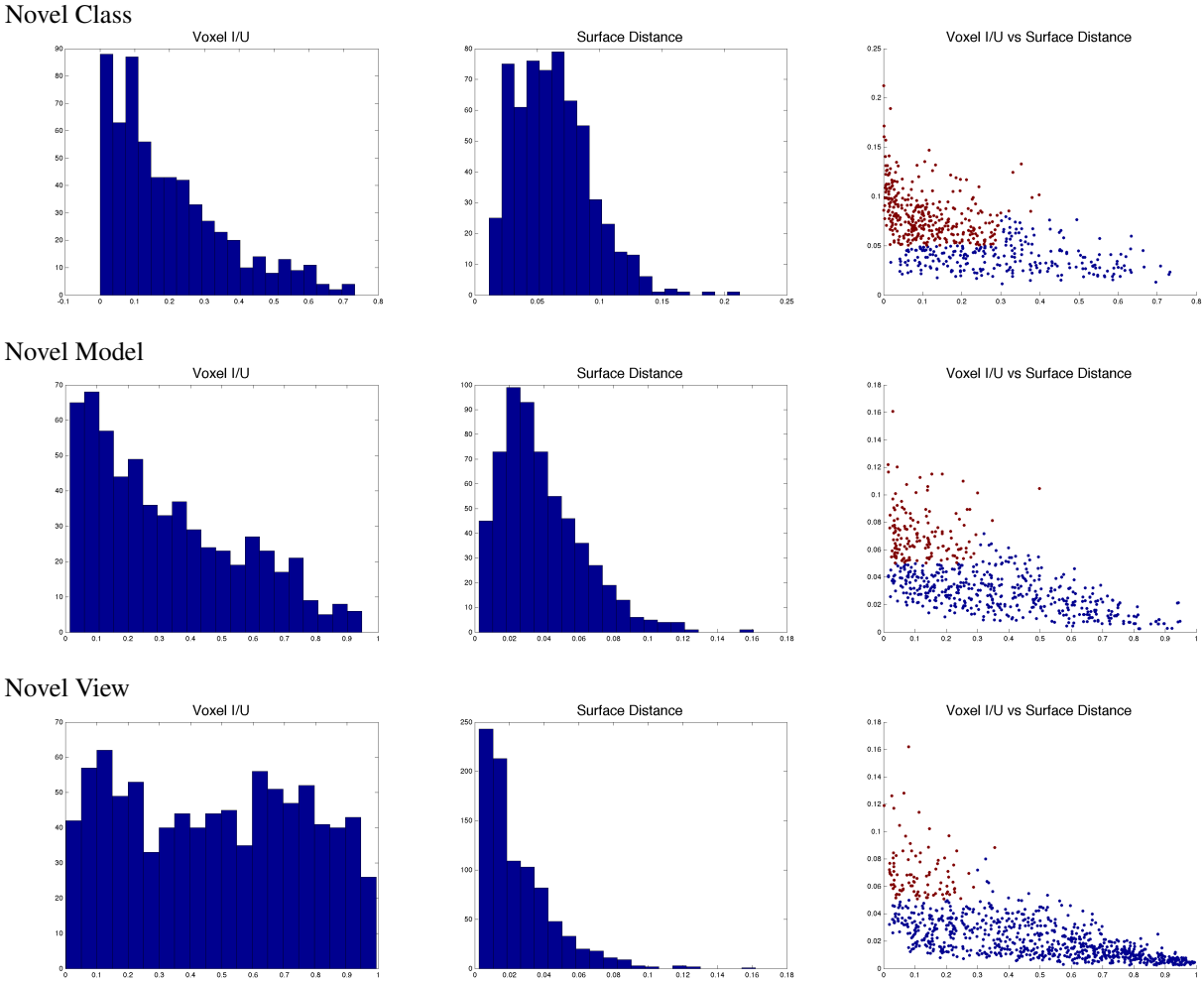


Figure 7: A visualization of the quantity of good matches. From left to right: voxel histogram of matches, surface histogram of matches, scatter plot of voxel scores vs surface distance scores (red indicates a poor match, blue indicates a reasonable match, cutoffs chosen empirically using inferences from Figure 6.

thin-plate-spline transform. These measures use only the retrieved exemplar to predict the query object shape. Finally, "Reconstruct Full" is our full proposed method that combines features from the retrieved exemplar and observed depth points to predict complete object shape. Our full method outperforms "Reconstruct Depth" substantially for the "novel model" and "novel view" cases and performs similarly in the "novel class" case. Thus, as expected, the retrieved meshes are more helpful when examples are available of the same model or a model from a similar class. See Figure 8 for examples of reconstructions.

Figure 9 illustrates the benefit of using deformation and symmetry reconstruction over simply returning the matched mesh or only using depth reconstruction. Reconstruction nearly always improves the fit of the matched exemplar, and in the cases of novel model and novel view also achieves better reconstructions than simple depth reconstruction. However novel class reconstructions can perform poorly due to poor exemplar matches and may worsen the fit of the model.

Voxel I/U		Reflect Depth	Reconstruct Depth	Match	Aligned	Deformed	Reconstruct Full
Novel Class	Mean	0.071	0.432	0.120	0.206	0.219	0.426
	Median	0.047	0.428	0.151	0.159	0.162	0.422
Novel Model	Mean	0.083	0.460	0.325	0.390	0.408	0.511
	Median	0.055	0.463	0.264	0.365	0.385	0.517
Novel View	Mean	0.089	0.453	0.478	0.590	0.599	0.581
	Median	0.059	0.452	0.482	0.652	0.651	0.607
Surface Dist		Reflect Depth	Reconstruct Depth	Match	Aligned	Deformed	Reconstruct Full
Novel Class	Mean	0.166	0.027	0.063	0.07	0.063	0.030
	Median	0.159	0.022	0.060	0.066	0.057	0.025
Novel Model	Mean	0.153	0.029	0.039	0.041	0.037	0.024
	Median	0.141	0.024	0.033	0.034	0.029	0.020
Novel View	Mean	0.1495	0.030	0.025	0.023	0.021	0.021
	Median	0.138	0.025	0.018	0.013	0.012	0.017

Table 3: Our method achieves drastic improvements over the baseline depth reflection, and when we have a well matching mesh (Novel model and Novel View), retrieving a match allows us to make improvements over just using our depthmap based reconstruction method.

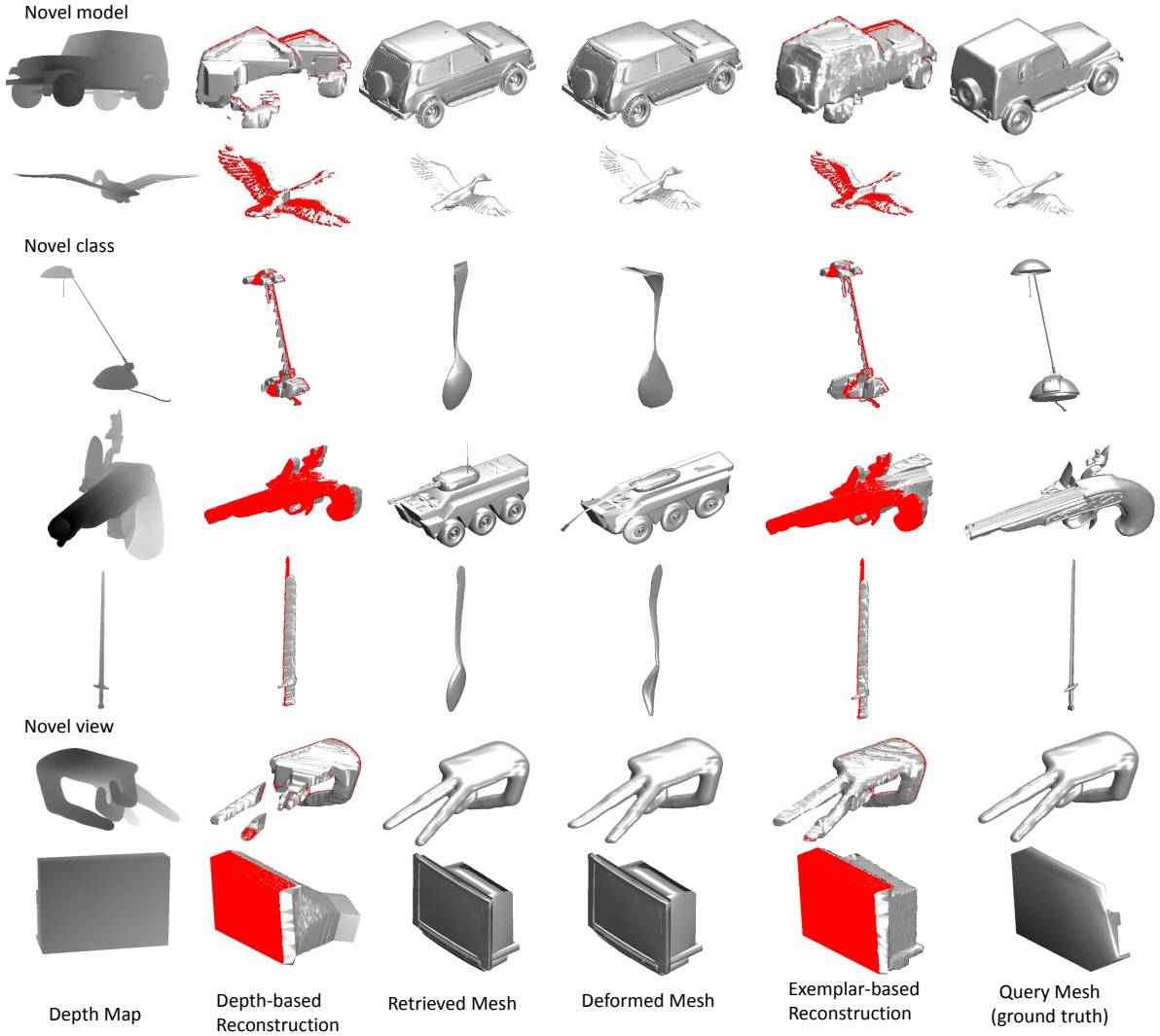


Figure 8: Examples of completed shapes. From left to right: input depth image, reconstruction based on depth image only, retrieved exemplar mesh, deformed exemplar, reconstruction based on exemplar and input depth, and ground truth. Red dots show the positions of the observed depth points. In each image after the first column, we show a new view so that unobserved parts of the object are visible.

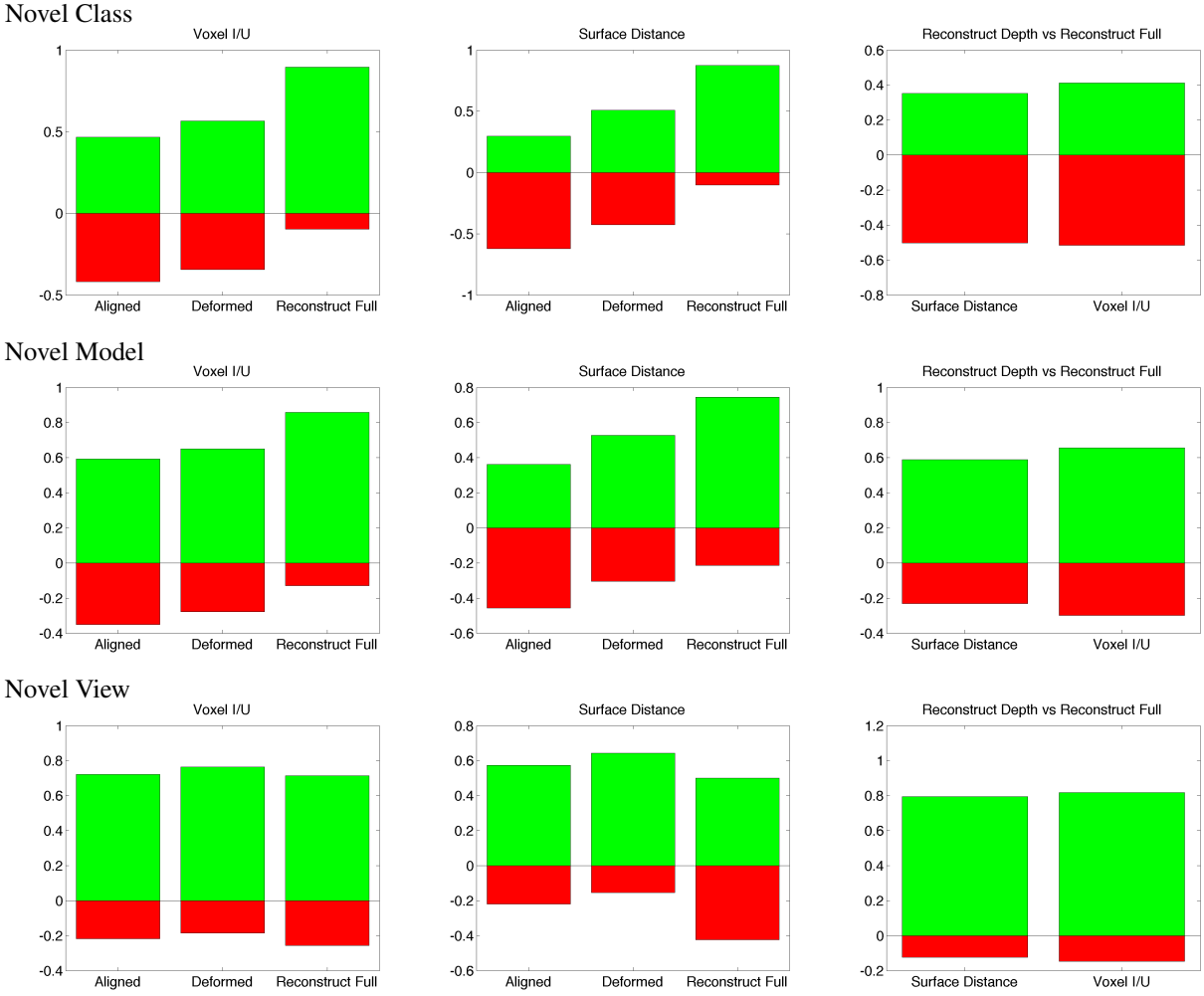


Figure 9: The two left columns show the percentage of mesh fittings that improve for each step of reconstruction compared to simply returning the matched model. The right column shows the percentage of reconstruct full meshes that improve fitting over reconstruct depth.

CHAPTER 4

DISCUSSION

Our reconstruction outperforms the naive baseline of "Reflect Depth" in all experiments and outperforms "Reconstruct Depth" in both novel view and novel model. We perform similarly to "Reconstruct Depth" for novel class. It is clear that a retrieved exemplar improves reconstruction if it closely matches the query object. It is challenging to complete a model of a novel class. However if we find a close enough fit, the symmetry constraints allow us to perform impressive deformations of unrelated models to create a reasonable and complete mesh. For example in Figure 8 we are able to deform the cannon and body of a tank to more accurately fit the barrel and trigger of a hand gun. However poor exemplar matches may weaken the reconstruction. Our matches are reasonable, but not optimal as shown in Table 2, we are still significantly below the optimal voxel and optimal surface matches.

The qualitative information in Figure 6 shows that surface distance is a strong indicator of mesh agreement and that voxel intersection over union can be a strong indicator as well, but only if the models are properly aligned. Thus, a next step could be to train a random forest designed to minimize the surface entropy rather than the voxel entropy. This would likely achieve better matches for complex objects with detailed surfaces, but may perform worse for objects with large and simple bodies. Another method to improve our matching approach could be to perform rigid alignment prior to selecting from potential matches, or perhaps even before training as human error is prevalent in some manually aligned models.

This paper focused on 3d model completion from a single depth observation on a synthetic dataset. The benefit of using a synthetic dataset is that we have high quality depth images with minimal noise to use for training and queries. It also simplifies the matters of different cameras, scales, and frames of reference since we keep these uniform. The main disadvantage of using a synthetic dataset is that our results may not reflect the system's ability to handle the noise present in natural depth images and point clouds, along with the poor resolution and dramatic changes in scale.

Two future goals are reconstructions from natural depth images and reconstructions from 2D RGB images instead of 2.5D depth images. As shown in Table 2, the random forest with depth features did not greatly improve matching results compared to the random forest with silhouette features only. Thus we could train a random forest with entropy and silhouette features only and expect similar results. As for reconstruction we would only be able to retrieve the exemplar and do minimal alignment using contours. Thus results would be on par with "Match" from Table 3.

CHAPTER 5

CONCLUSION

We proposed a method for reconstructing a complete 3D model from a single depth observation. Our method involves retrieving a similar 3D mesh exemplar, based on depth image matching, deforming the exemplar mesh to better fit the observed depth points, and predicting the complete shape of the target object based on both the deformed exemplar and observed depths. We propose a new dataset to evaluate shape reconstruction based on the SHREC'12 dataset. Our experiments indicate that the quality of the reconstruction depends strongly on the similarity of available exemplar meshes, but that good reconstructions are sometimes possible, even when a similar exemplar is not available. We hope that by providing a benchmark dataset, we will encourage future research in object shape reconstruction.

REFERENCES

[1] A. A. Amini, Y. Chen, R. W. Curwen, V. Mani, and J. Sun. Coupled b-snake grids and constrained thin-plate splines for analysis of 2-d tissue deformations from tagged mri. *Medical Imaging, IEEE Transactions on*, 17(3):344–356, 1998. 2

[2] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014. 1, 2

[3] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992. 2

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 2

[5] T. J. Cashman and A. W. Fitzgibbon. What shape are dolphins? building 3d morphable models from 2d images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):232–244, 2013. 1

[6] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or. 3-sweep: extracting editable objects from a single photo. *ACM Transactions on Graphics (TOG)*, 32(6):195, 2013. 2

[7] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1841–1848. IEEE, 2013. 4

[8] M. Finch, J. Snyder, and H. Hoppe. Freeform vector graphics with controlled thin-plate splines. In *ACM Transactions on Graphics (TOG)*, volume 30, page 166. ACM, 2011. 2

[9] H. Fu and G. Qiu. Fast semantic image retrieval based on random forest. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 909–912. ACM, 2012. 4

[10] R. Gal, O. Sorkine, N. J. Mitra, and D. Cohen-Or. iwires: an analyze-and-edit approach to shape manipulation. *ACM Transactions on Graphics (TOG)*, 28(3):33, 2009. 1

[11] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen. Data-driven grasping with partial sensor data. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1278–1283. IEEE, 2009. 2

[12] B. K. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A*, 5(7):1127–1135, 1988. 5

[13] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3d freeform design. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 2

[14] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999. 2

[15] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999. 5

[16] N. Kholgade, T. Simon, A. Efros, and Y. Sheikh. 3d object manipulation in a single photograph using stock 3d models. *ACM Transactions on Computer Graphics*, 33(4), 2014. 2

[17] C. Kurz, X. Wu, M. Wand, T. Thormhlen, P. Kohli, and H.-P. Seidel. Symmetry-aware template deformation and fitting. *Computer Graphics Forum*, 33(6):205–219, 2014. 2

[18] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. 4

[19] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1482–1489. IEEE, 2005. 5

[20] B. Li, A. Godil, M. Aono, X. Bai, T. Furuya, L. Li, R. J. López-Sastre, H. Johan, R. Ohbuchi, C. Redondo-Cabrera, A. Tatsuma, T. Yanagimachi, and S. Zhang. Shrec'12 track: Generic 3d shape retrieval. In *3DOR*, pages 119–126, 2012. 8

[21] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing ikea objects: Fine pose estimation. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2992–2999. IEEE, 2013. 1, 2

[22] J. Liu. Efficient recognition of rotationally symmetric surfaces and straight homogeneous yeneralized cylinders jane liu*, joe mundy*, david forsyth*, andrew zisserman and charlie rothwell(*) te center for research and development, 1 river rd, schenectady, ny 12345.(*) department of computer science, university of iowa, iowa city, ia 52242. 1993. 2

[23] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (TOG)*, 25(3):560–568, 2006. 2

[24] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (TOG)*, 25(3):560–568, 2006. 5

[25] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno. Salient local visual features for shape-based 3d model retrieval. In *Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on*, pages 93–102. IEEE, 2008. 2

[26] M. Prasad and A. Fitzgibbon. Single view reconstruction of curved surfaces. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1345–1354. IEEE, 2006. 2

[27] K. Rohr, H. S. Stiehl, R. Sprengel, T. M. Buzug, J. Weese, and M. Kuhn. Landmark-based elastic registration using approximating thin-plate splines. *Medical Imaging, IEEE Transactions on*, 20(6):526–534, 2001. 5, 6

[28] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, 2007. 1, 2

[29] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM, 2004. 1

[30] Y. Wang, J. Feng, Z. Wu, J. Wang, and S.-F. Chang. From low-cost depth sensors to cad: Cross-domain 3d shape retrieval via regression tree fields. In *Computer Vision–ECCV 2014*, pages 489–504. Springer, 2014. 2

[31] W. Wohlkinger and M. Vincze. Shape-based depth image to 3d model matching and classification with inter-view similarity. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4865–4870. IEEE, 2011. 2

[32] Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao. 3d shapenets for 2.5 d object recognition and next-best-view prediction. *arXiv preprint arXiv:1406.5670*, 2014. 2

[33] K. Xu, H. Zheng, H. Zhang, D. Cohen-Or, L. Liu, and Y. Xiong. Photo-inspired model-driven 3d object modeling. In *ACM Transactions on Graphics (TOG)*, volume 30, page 80. ACM, 2011. 2

[34] Y. Zheng, X. Chen, M.-M. Cheng, K. Zhou, S.-M. Hu, and N. J. Mitra. Interactive images: cuboid proxies for smart image manipulation. *ACM Trans. Graph.*, 31(4):99, 2012. 1, 2

[35] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3D Object Shape from One Depth Image. *In Submission CVPR 2015*, 2014. 1